Labs

There are 4 exercises in this lab, of which the last exercise is "extra credit". Here is a brief summary of the tasks you will perform in each exercise; more detailed instructions follow later:

1. Refactoring code to use new array features
2. Copying within an array
3. Finding an element in an array
4. (Extra credit) Using typed arrays with the File API

Exercise 1: Refactoring code to use new array features

Refactor the code to make use of ES6 array features where possible. For example:

- Use for-of loops rather than for-in loops. Try having a go at using the version of for-of that gives the index and value of every element, and call console.log() on each iteration to display indexes/values on the console.
- Also try out forEach() as an alternative iteration mechanism somewhere in the code.

Exercise 2: Copying within an array

In the All Product Suggestions panel, add a Repeat button to repeat the last product in the array. The idea is that if the array contained [A,B,C] initially, then it'll contain [A,B,C,C] afterwards.

Here's what you need to do:

- Increase the size of the array by 1 element, via push().
- Clone the element that used to be at the end position to the new slot at the end of the
  array. Use copyWithin() to do this.

Exercise 3: Finding an element in an array

In the Matching Product Suggestions panel, add a Find First button to find the first product matching the text entered by the user. Use find() to do the work. If a match is found, display it in the panel, otherwise display an error message (e.g. in an alert).

Exercise 4 (If time permits): Using typed arrays with the File API

HTML5 has a standardized File API that allows a web app to interact with files on the local file
system. You'll explore this API in this exercise.

In the student folder, go to the ReadingFiles folder and open index.html in a browser.

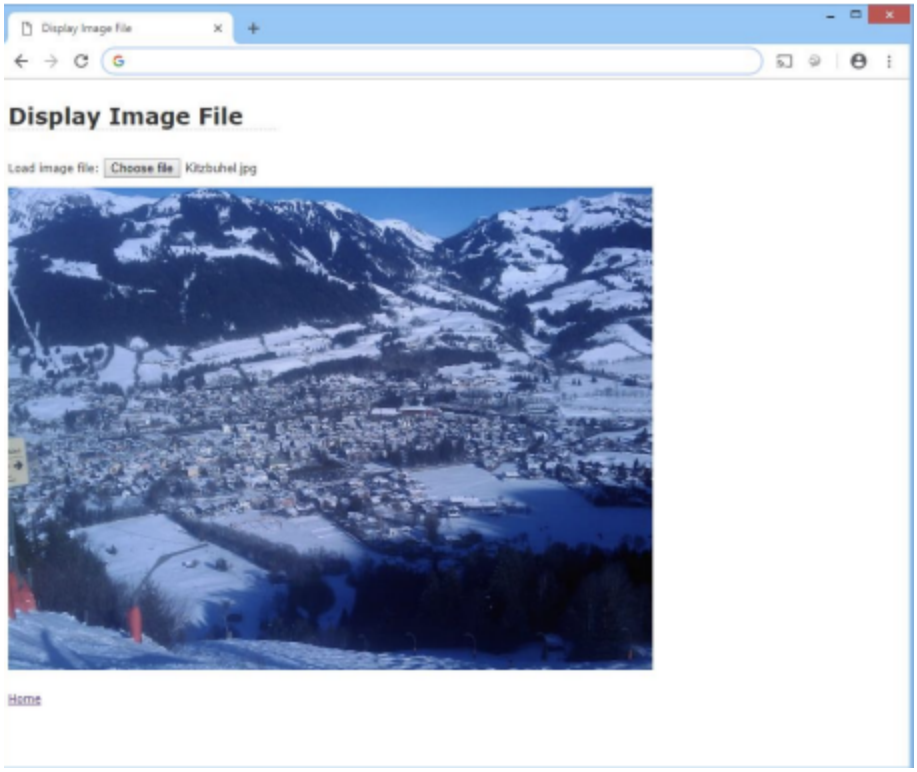
There are 3 hyperlinks that take you to separate pages, to read a file in 3 different ways:

- As an image file (already implemented)
- As a text file (already implemented)
- As a binary file (you will implement this)

Click the Display image file link first, which takes you to displayImageFile.html. The web page has a Choose file button – this is actually an HTML5 element. When you click it, a file chooser dialog box appears. Choose an image file in the files subfolder, e.g. Kitzbuhel.jpg and click Open. This generates a change event, which is handled in onLoadImageFile() in es6scripts/processImageFile.js.

This function does the following tasks:

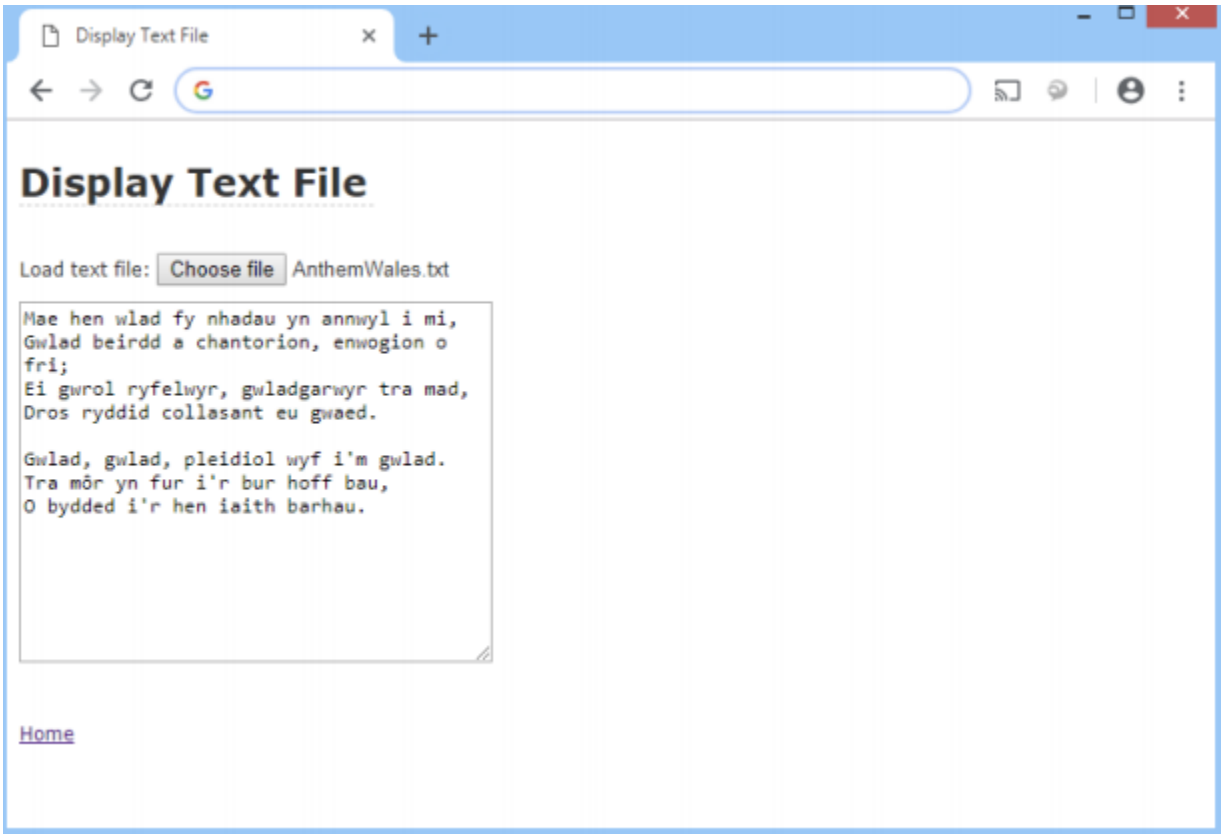
- It checks the user selected an image file, and gets the File object representing the file.
- It creates a FileReader object to read the file contents. Specifically, FileReader has a readAsDataURL() method that reads a file asynchronously and exposes the contents as a URL, which can be assigned to the 'src' attribute on an element. The net effect is that the web page displays the image you selected.



When you're happy with this, click the Home link to return to the home page.

Back on the home page, click Display text file which takes you to displayTextFile.html. This web page is similar to the previous one, except it reads a text file via the readAsText() method on FileReader.

To see how it works, see onLoadTextFile() in es6scripts/processTextFile.js. The net effect is that the web page displays the text file you selected, e.g. here's the content of the AnthemWales.txt file:



When you've had a play with this, click Home to return to the home page.